Vehicle Speed Estimation with Non-stationary Camera

Gaofeng Su

Abstract

Vehicle speed estimation plays a vital role in traffic management, data analysis, and Intelligent Transportation Systems (ITS). While traditional speed sensing systems, such as radar and laser-based technologies, offer high accuracy, their hardware costs remain a significant barrier. In contrast, vision-based approaches are gaining attention due to their relatively simple and cost-effective hardware requirements. However, most existing vision-based methods rely on stationary cameras, limiting their flexibility and the scalability of vehicle speed monitoring networks.

This paper proposes a simple, but effective homography-based approach for vehicle speed estimation using a non-stationary platform. The method utilizes vehicle keypoints detection and computes the homography matrix. By warping the 2D velocity vector obtained from the optical flow method, the actual vehicle speed is estimated with high accuracy. This approach offers a flexible and cost-efficient solution for vehicle speed monitoring.

1. Introduction

Vehicle speed monitoring plays a crucial role in traffic management, as it can improve road safety. According to the San Francisco Municipal Transportation Agency (SFMTA), San Francisco is introducing the Speed Safety Cameras Program to enhance road safety [1]. Additionally, with the growing interest in Intelligent Transportation Systems (ITS), on-road vehicle speed estimation can provide valuable traffic speed data for further analysis by ITS.

Traditional speed camera monitoring systems typically rely on expensive hardware such as radar, lidar, and vision cameras, which makes scaling a speed safety camera network challenging. With modern CPUs becoming more powerful and the advancement of computer vision technology, lots of research has been conducted on vision-based vehicle speed estimation methods [2]. However, most of these vision-based method are designed to works on stationary speed cameras. Relying on a stationary camera reduces the flexibility and scalability of the vehicle speed monitoring networks since the cost of the speed camera system is still relatively high.

One idea is to use non-stationary platforms such as phone cameras, which are low-cost and easy to set up to capture videos, to perform vehicle speed estimation. This not only introduces a new form of speed-sensing modality but also makes this sensing network more flexible and scalable.

Traditionally, use the camera to perform vehicle speed estimation, one key component is the camera parameters (i.e., the intrinsic and extrinsic matrix)[2]. For stationary cameras, this parameter can be calibrated offline, but this is tricky for non-stationary cameras as the extrinsic matrix changes over time and is hard to calibrate in real-time. Research has been done using sophisticated pipelines such as reconstructing 3D bounding boxes of vehicles to estimate speed. This paper proposed a simple method based on homography and warping technique but could yield relatively high accuracy. The paper is arranged as follows: Section 2 explains the methods and processing pipeline, section 3 will briefly describe the experiment setup section 4 is about results discussion, and Section 5 is the conclusion and future work.

2. Methdology

2.1. Relating Vehicle Speed with Homography

The key to the proposed method is that two images can be related by homography if the object/scene being captured is on a single plane even if the camera is not in the same center of projection (COP). Noticed that this observation still holds even though the extrinsic matrix of the camera is changing.

This is commonly used in many stationary camera calibration methods, people either use known planar object to calibrate the camera, or assume the road is a plane and use known road features such as traffic lines, to calibrate the camera. The challenge here is that for a scenario like a hand-held camera, it is quite hard to have a clear view of the road surface. Moreover, this requires good maintenance of the road so you can always see these road features, which might not be realistic.

However, we can approach it from a different angle,

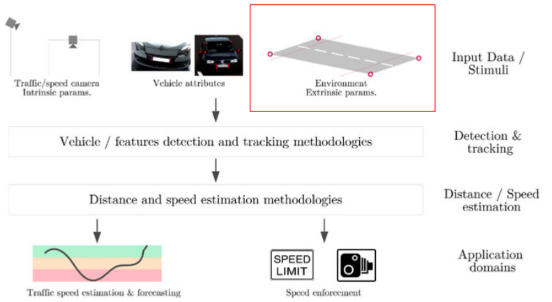


Figure 1. Common speed estimation pipeline, image adapted from [2]

.

Figure 2. Extract road features to find parallel lines, horizon etc. image adapted from [6]

.

other than the road surface, we can approximate the vehicle's side as a single plane. And since we are estimating the vehicle speed, we must see it from our view, which means we are more likely to observe the side of the vehicle.

Now we can use homography and warping to compute the velocity obtained from optical flow. But there is is assumption: the vehicle's speed must aligned with the vehicle's side plane. Below is a less formal explanation:

Denoted a vehicle keypoint on image k as s_k, the same keypoint on image $k+1$ as s_{k+1}, let the matching keypoint be s, and the computed homography matrix to be H_k and H_{k+1} respectively. We have:

$$w_k s = H_k s_k \tag{1}$$

$$w_{k+1} s = H_{k+1} s_{k+1} \tag{2}$$

The speed vector from the optical flow is:

$$v = \frac{1}{dt}(s_{k+1} - s_k) = \frac{1}{dt}(H_{k+1}^{-1} w_{k+1} s - H_k^{-1} w_k s) \tag{3}$$

Since the speed vector and the homography matrix are computed independently, this means there is a constraint on the resulting scaling factor w_{k+1} and w_k. Since the scaling factor controls the point's depth, and the optical flow is tracking the surface of the vehicle's speed, you can interpret this as the actual speed must aligned with the vehicle side plane to use the homography to warp the speed vector from optical flow.

2.2. System Pipeline

Based on this idea, the following pipeline is proposed, see Figure 3. The input to the system is a video stream and a pre-defined vehicle keypoint template with known dimensions.

The first step is to compute the dense optical flow map, we used OpenCV's dense optical flow implementation of Gunnar Farneback's algorithm [5]. Meanwhile, a keypoint detector detects the vehicles and the keypoints in the current image and matches the keypoints with the pre-defined template to obtain only the side plane keypoints. If we have more than four keypoints, we can compute the homography matrix and use it to transform the speed vector on these keypoints. In an ideal case, the transformed speed vector should all pointing horizontally (i.e. 0 on the verticle axis). After obtaining all observed keypoints' speeds, the final speed will be computed as the average of the keypoint speeds.

2.3. About Vehicle Template and Detection Model

Machine-learning-based keypoint detection model proven to be robust to detect vehicle keypoints with "semantic meaning" such as "the center of front/rear wheels", corner of the windows. However, it is challenging to have one universal model to include all vehicles and with realistic mechanical dimensions. In this paper, we use the openpifpaf car keypoint detection model [3], which uses a 66-keypoints sedan model, see Figure 4.

3. Experiment

To test the proposed method and the pipeline, we collected four hand-held camera footage and performed speed estimation offline. The videos are recorded in 60 fps with a resolution of 2160 × 3840. To speed up the process it was resized to 540x960. Each footage lasts around 3 seconds. We also used a speed gun to measure the actual vehicle speed as the ground truth. Figure 5 illustrates the estimation and the transformed speed vector of vehicle 1.

Belows are the comparison between the estimated vehicle speed from hand-held camera footage and the speed gun measurments.

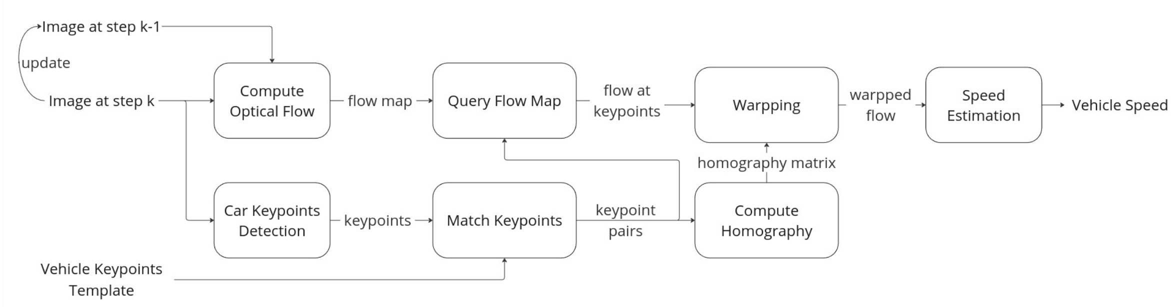


Figure 3. Proposed system pipeline

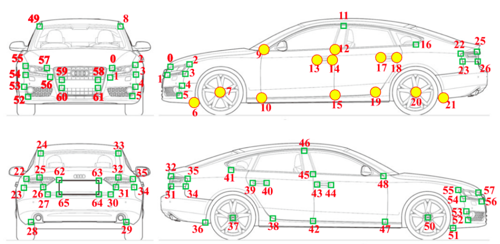


Figure 4. 66-keypoints sedan model, image adapted from [4] . The target keypoint is highlighted in yellow.

4. Result and Disccusion

The three experiments show that the proposed method has a relatively high accuracy. The estimation of vehicle 3 seems inaccurate at first glance, but after careful analysis of the footage, we found out that vehicle 3 in this footage slowing down due to the traffic signal turning red. Therefore, although we could not provide a continuous measured ground truth, this result is reasonable in this case. Another observation is that, after some time, the estimated speed dramatically changed and yielded incorrect results. This is because as the car drove away from the camera, the viewing angle to the side of the vehicle became very narrow, and it was hard to detect many features. The features it detected are almost colinear, which causes singularity or near singularity issues when solving the homography matrix, as illustrated in Figure **??**. This is one limitation of the proposed methods. But this might be able to mitigated by adding more keypoints (e.g. more keypoints on wheels).

5. Conclusion and Future Work

This paper proposed and simple and effective method to estimate vehicle speed on a non-stationary platform by treating the side of the vehicle as a single plane, matching keypoints to a pre-defined template, and transforming the speed vector from optical flow. From the hand-held camera experiments, we can see that the proposed method can accurately measure the speed while ”good features” of the vehicle are detected. One limitation is that as the vehicle moves away from the camera and the viewing angle to the side of the vehicle becomes narrow, the estimation becomes unreliable. This might be able to be mitigated by adding more keypoints to the vehicle model.

Currently, we are using a pre-trained sedan keypoint model, however, this can be improved by training keypoint detection model that is tailored to a specific type of vehicle. So that we can not only detect and estimate the speed of sedans but also vehicles like SUVs, truck, and bus. Also customized keypoints model by adding more keypoints to places like wheels might make the section more robust against poor-viewing angle.

References

[1] San Francisco Municipal Transportation Agency: ”Why we’re introducing speed safety cameras, a first for California,” accessed Nov 21, 2024. [Online]. Available: https://www.sfmta.com/blog/why-were-introducing-speed-safety-cameras- first-california

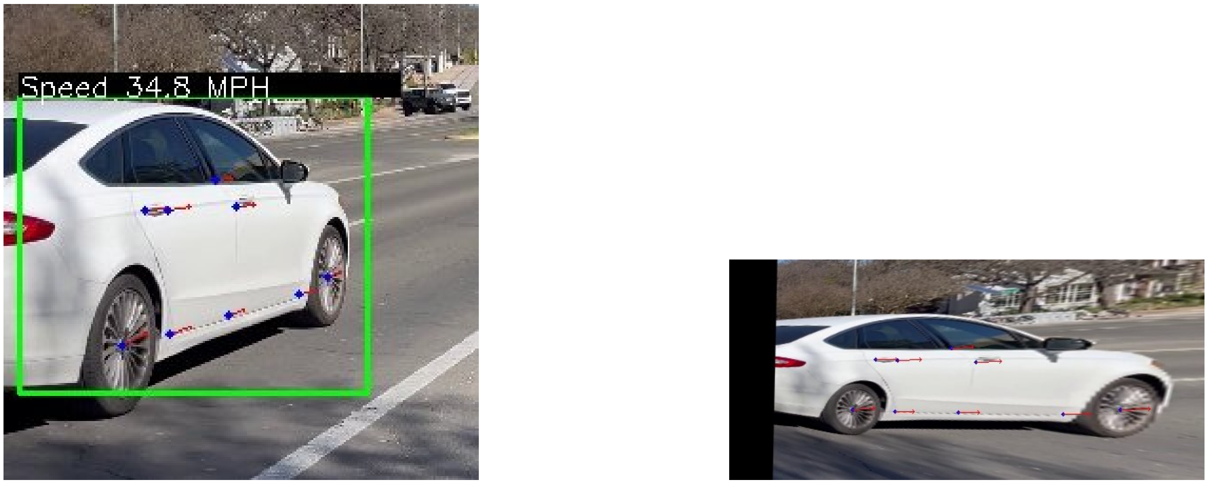
[2] David Fernández Llorca et al.: ”Vision-based vehicle speed estimation: A survey,” 2021.

[3] OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association, Sven Kreiss et al. (2021)

[4] ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving, Xinbin Song et al. (2019)

[5] OpenCV, https://docs.opencv.org/3.4/dc/d6b/group$_{video_{track.htmlga5d10}}$

[6] Automatic Roadside Camera Calibration with Transformers, Yong Li et al., 2023



(a) Estimated speed.

(b) Transformed image and the speed vector.

Figure 5. Speed estimation example of vehicle 1. Speed vectors are marked by red arrow, and keypoints are marked in blue circles.

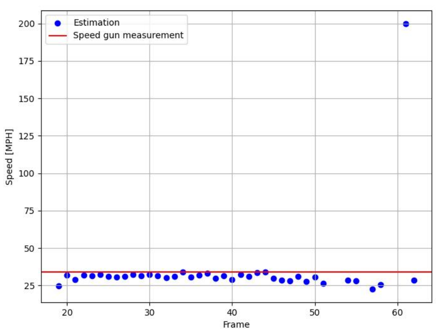


Figure 6. Estimated speed of vehicle 1 in footage recorded by a hand-held camera.

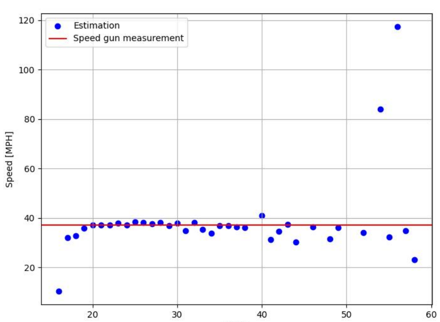


Figure 7. Estimated speed of vehicle 2 in footage recorded by a hand-held camera.

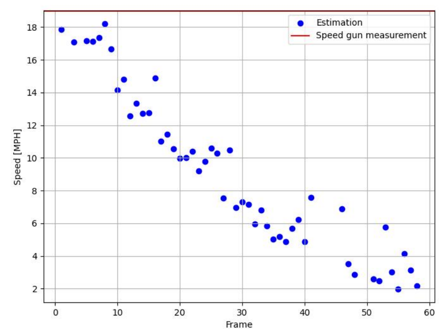


Figure 8. Estimated speed of vehicle 3 in footage recorded by a hand-held camera.

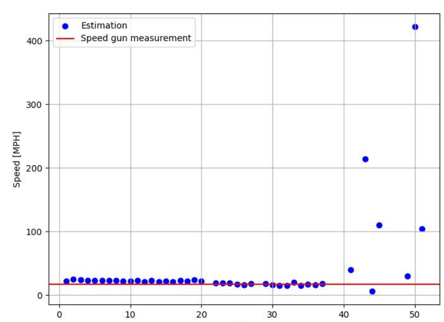


Figure 9. Estimated speed of vehicle 4 in footage recorded by a hand-held camera.

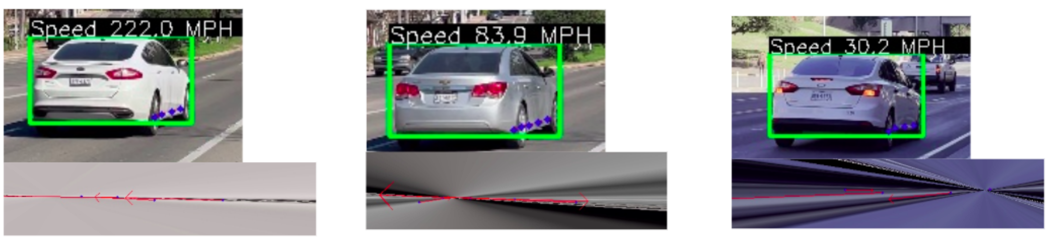


Figure 10. Poor viewing angle makes the detector only detect keypoints that are co-linear and introduce singularity issues of the homography matrix. The top row of images, shows the detected keypoints, as you can see they are forming a straight line. The bottom row shows the transformed image, and you can see the speed vector is randomly oriented.